CHAPTER-3

Abstract Mathematical Model for Database Schema Using Reference Graph
CHAPTER-3

ABSTRACT MATHEMATICAL MODEL FOR DATABASE SCHEMA USING REFERENCE GRAPH **

3.1 INTRODUCTION

It is found that Entity-Relationship diagram plays a vital role in structured analysis and conceptual modelling. Entity-relationship (ER) model is a commonly used abstract representation of database structure. In building a relational database the first step is the creation of an ER model. An ER model is a representation of a database structure. It is not about the contents of the database. There are many classification of E-R model depending of their definition and representation [26], and a comparison of various models is discussed in by Il-Y. Song, et al.[27]. They have discussed about how E-R models are developed to define database. These are basically graph based data model. The entities in a system and their relationships are modelled by using graph based models. Graph based approaches are widely used in database management, data retrieval and data mining. The E-R models represent the entities and their relationship only. P. S. Dhabe, et al. [28] has proposed a new model called ‘Articulated Entity Relationship’ (AER) as an extension of Entity Relationship (ER) diagram to accommodate the Functional Dependency (FD) information. Updating the existing attributes may lead to inconsistent description of FD information and attribute description of entities. [28]. D.Heckerman, et al. [29], proposed a model that incorporated the probabilistic relationship into E-R models. A. Sarkar [30] in his paper proposed a graph based approach for modelling of irregular, heterogeneous, hierarchical

** This chapter has been published as a research paper in the journal “International Journal of Engineering Research and Development”(IJERD), Vol.- 6, Issue- 4,March, 2013, pp. 51-60.
and non-hierarchical semi-structured data at the conceptual level in object oriented paradigm.

Many graph based approaches are being used in knowledge and data engineering. Y. Yang [31], in his paper, proposed a new recommendation technique based on reference graph which can recommend research paper in an effective manner. The reference graph based recommendation technique is more effective [31]. M. Graves et al. [33] proposed a graph representation of Genome database. M. Tavana, et al. [34] has proposed an algorithm for re-clustering of E-R models. They use a method for clustering of entities to form small interrelated modules of the E-R diagram consisting of very large number of entities and relationships. R. Angels et al. [139], in their paper presented a survey to present the work that has been conducted in the area of graph database modelling. There are many automated tools for designing database schema. L. Simasatitkul et al.; [36] in their paper, proposed a tool for generating database schema from EER diagram, written in XML format. The goals of this tool are transforming an EER diagram to relational database schema and to specify relational database constraints in the relations.

In this chapter, a simple method for designing database schema is presented which will help the students, designers and software developers. In the new approach, unlike E-R model, the reference graph model can describe the references occurred among the database tables or among various databases of diverse platforms more precisely and clearly. The reference graphs, unlike the schema diagrams, do not show the details about the attributes of the entities, rather it gives the references available among the entities and the attributes through which these references are made. Reference graphs model, derived from the E-R model or from the intermediate graph,
directly developed by analyzing the context system, is used to develop an abstract mathematical model of the database schema, from which the Database Requirements Specifications (DRS) is developed. The DRS can be used as a design document in RDBMS and OODBMS paradigm.

3.2 OBJECTIVE

Representing the E-R model as a graph is very helpful in database design and maintenance. If the structure of a table is to be changed to meet the future requirements (adding new attributes/ removing existing attributes), the number of out-degree and the number of in-degree of a node representing a table will give the number and names of the tables that may be affected due to the changes to be made. In this chapter, a comprehensive method for analysis and design of database schema is proposed which can be used as a design document in RDBMS and OODBMS paradigm.

3.3 REFERENCE GRAPH

The E-R model can be represented as a graph \( G = \{V, E\} \), called reference graph, where \( V \) is the set of entities and \( E \) is the set of relationships. The relationships are represented as directed graphs. The set \( V \) also includes the \( m: n \) relationships. There is a set, \( K_{SYS} \), the key set, which contains the primary keys of all sets in the system. A set representing an entity or a relationship in the set \( V \) contains two sub sets - the set of identifying attributes (keys), \( K \) and the set of describing attributes, \( D \).

3.3.1 Unary Relationship

In unary relationship same one entity is associated with itself. It is very important to specify the role of the participants in the relationship. The reference graph for unary relationship is shown in fig. 3.1.
3.3.2 Binary Relationship

The n:1 relationship is represented by an edge from n-side to the 1-side. For example, an edge from STUDENT to BRANCH is drawn as shown in fig. 3.2(b). For a 1:1 relationship an edge is drawn from one entity to the other as shown in the fig. 3.2 (a). For each m:n relationship a new vertex is created. For example, a new vertex SCORE for the <TAKE> relationship in the reference graph is created as shown in fig. 3.2 (c).

Figure 3.2: Binary Relationships and their Reference Graphs
3.3.3 Ternary Relationship

A ternary relationship and its corresponding reference graph are shown in fig. 3.3 (a) and fig. 3.3 (b) respectively.

![Ternary Relationship Diagram]

Figure 3.3: Ternary Relationship and its Reference Graph

3.3.4 Specializations and Aggregations

For specialization, in the reference graph a separate set for each specialized entity is created assuming that there is a reference from the specialized entity to the generalized entity as shown in fig. 3.4 (a). For mapping of specialization types the concepts discussed in by Lisa Simasatitkul et al, [36] are used. A relation is created for each subclass and all the simple attributes of subclass are also created. The primary key of subclass is derived from primary key of super class.

For aggregation, create a set for the relationship between the sets in the aggregation and this new set can be considered as a regular set in the system. An example is shown in fig. 3.4(b).
In an E-R graph, also called as Reference Graph, the vertex is nothing but a database table and the edges are nothing but the associations among the tables. In this context the following theorems are proposed.

3.4 THEOREMS

3.4.1 Theorem 6: In a reference graph, the out-degree of a vertex represents the number of tables (vertices) it has referenced. [The edge names represent the fields or attributes through which it has referred to other tables (vertices)].

Proof: [The vertices of the reference graph are nothing but sets of attribute types. For each reference graph there is a key set, K]
Let $A$ and $B$ are two sets and

$$A = \{b_1, a_1, a_2, a_3, ..., a_n\}$$

$$B = \{b_1, b_2, b_3, ..., b_m\}$$

Now, $A \cap B = \{b_i\} \ 1 \leq I \leq m \quad \text{(3.1)}$.

Equation (1) means that there is a reference from $A$ to $B$ through $b_i$, $(b_i \in B) \& (b_i \in K)$, and it is graphically represented in the reference graph as shown in fig. 3.5(a).

Now, let us assume that there exists a set $A$ and a set of set $X=\{X_1, X_2, X_3, ..., X_k\}$ in the system. By equation (1), if $A \cap X_i = \{x_{ij}\}, \ 1 \leq j \leq m$, then there must be an outgoing edge from $A$ to the set $X_1$ in the reference graph of the system. If there is at least one reference from $A$ to any set in $X$, then the following inequality must hold.

$$\bigcup_{i=1}^{k} (A \cap X_i) \neq \varnothing \quad \text{(3.2)}$$

By equations (1) and (2) it is found that, if $|\bigcup_{i=1}^{k} (A \cap X_i)| = v, v>0$ then there are $v$ number of references from $A$ to set $X$ which is represented by $v$ number of outgoing edges from $A$ in the reference graph.

**Example:** In fig. 3.2(b) the vertex SCORE has referenced to the vertex STUDENT by the attribute ‘rollno’ and to the vertex SUBJECT by ‘sub-code’. This means that the table has reference to two tables- STUDENT and SUBJECT.

3.4.2 **Theorem 7:** In a reference graph, the in-degree of a vertex represents the number of tables (vertices) where the attributes, represented by the edges, of the table (vertex) are foreign keys. [Proof is similar to Theorem 3.4.1]
Example: In fig. 3.2(a) the attribute BR-CODE of table (vertex) BRANCH is a foreign key in the table (vertex) STUDENT.

3.4.3 Theorem 8: The reference graphs without self loop are always acyclic.

An edge is drawn from a node to another node if there is an n: 1 or 1: 1 mapping. In fig. 3.6(a), the following mappings: (A<n: 1>B<m: 1>C<p: 1>D<q: 1>A) are shown.

Figure 3.6: (a) Cyclic Reference and (b) Acyclic Reference

To each B-value, many A-values are associated (e1), to each C-value; many B-values are associated (e2). So, to one C-value, (n×p = r) A-values are associated (edge e5). Similarly, to each D-value, many C-values are associated (e3). By edge e4, it is seen that to each A-value, many D-values are associated. But if the edges e5 and e3 are combined, the edge e6 is obtained, meaning that, for each D value u=r×q values are associated. Therefore, it can be stated that between D and A the mapping is many-to-many (u: m). But for many-to-many relationship a separate node is created with an edge to A and an edge to D, resulting a tree. Hence the graph cannot be cyclic and a reference graph as shown in fig. 3.6 (b) is obtained.

3.5 OUTLINE OF THE APPROACH

The proposed approach comprises of seven steps (see fig. 3.7). First, identify the entities and their attributes in the system. Second step is to identify the key attributes of the set. In the third step, create the sets for the entity sets and add the key attributes in
the key sets of the set. Fourth, identify the relationships among the entity sets and draw
the intermediate graph showing the cardinalities of the relationships. In the fifth step
convert the undirected edge of the intermediate graph by following the steps described
in section v. Draw the reference graph and develop the abstract model. Then, define
$K_{SYS}$ and represent the reference graph as a matrix. Finally develop the mathematical
model (DRS) for the database schema.

---

![Figure 3.7: Outline of the Approach](image)

---

Figure 3.7: Outline of the Approach
3.6 STEPS IN DEVELOPING THE ABSTRACT MODEL

Step 1: Draw a node for each entity set.

Step 2: Find out the relationships among the sets (nodes) and join them with an undirected edge mentioning the mapping cardinalities as shown in fig. 3.9.

[Up to this step, only the intermediate graph is obtained.]

Step 3: Replace the edges with n: 1 or 1: n and 1:1 cardinality with directed edge from n-side to 1-side and label the edge as shown in fig. 3.10 (k NODE-NAME). For an m: n edge or an m:n:p relationship, create a new set with an appropriate name, remove the undirected edge from the graph and draw directed edges from the new node to the nodes participating in the relationship. Label the edges accordingly. Define the key set for the newly created node.

Step 4: Develop an abstract mathematical model for the system. (As shown in fig. 3.11)

3.7 REPRESENTATION OF THE REFERENCE GRAPH

The nodes of the reference graph can be defined as sets. Each set has a set of identifying attributes and a set of describing attributes. The reference graph of E-R model can be represented as a matrix (other representation is also possible, like sparse matrix). The contents of the matrix are the attribute(s) through which the vertices are associated. For example, let us consider the reference graph of fig. 3.2(a) and fig. 3.2(b). The reference graphs in fig. 3.2 (a) and fig. 3.2 (b) are represented in the matrices given in Table-3.1 and Table-3.2 respectively as follows.

In case, if the reference are through composite attributes then the contents of the matrix are lists of attributes names.
### Table 3.1: Reference Matrix for fig. 3.2(b)

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>BRANCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>X</td>
</tr>
<tr>
<td>BRANCH</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 3.2: Reference Matrix for fig. 3.2(c)

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>STUDENT</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>STUDENT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SCORE</td>
<td>ROLLNO</td>
<td>SUBCODE</td>
</tr>
</tbody>
</table>

3.8 GENERATING DATABASE REQUIREMENT SPECIFICATIONS (DRS)

**Step 1:** Create the set $K_{SYS}$. Create a set for each entity in the E-R diagram. Add the identifying attributes in the key set of the set created for the entity and other attributes in the describing set. Add the primary key of the set to $K_{SYS}$.

**Step 2:** Search the reference graph to identify the nodes which have outdoing edges. Create a set for each node identified, if it not created in Step 1.

**Step 3:** For each node identified in Step 2, add the attribute(s) specified in the outgoing edge labels of this node in the appropriate set (key set or describing set).

By following the above steps for the reference graph in fig. 3.3(b), the specifications of the database for the system described by the reference graph is as follows.

*By Step 1, the following sets are obtained.*

$$K_{SYS} = \{ \{ROLL-NO\}, \{SUB-CODE\} \}$$
STUDENT = \{\{Roll-No\}, \{Name, Address\}\}

SUBJECT = \{\{Sub-Code\}, \{Sub-Name, Tot-marks\}\}

By Step 2 and Step 3, the following sets are obtained.

SCORE = \{\{Roll-No, Sub-Code\}, \{Sub-Name, Tot-marks\}\}

K_{SYS} = \{\{Roll-No\}, \{Sub-Code\}, \{Roll-No, Sub-Code\}\}

The complete database specifications are as follows.

DRS: STUDENT-SUBJECT

K_{SYS} = \{\{Roll-No\}, \{Sub-Code\}, \{Roll-No, Sub-Code\}\}

SET STUDENT = \{\{Roll-No\}, \{Name, Address\}\}

SET SUBJECT = \{\{Sub-Code\}, \{Sub-Name, Tot-marks\}\}

SET SCORE = \{\{Roll-No, Sub-Code\}, \{Sub-Name, Tot-marks\}\}

3.9 AN EXPLANATORY EXAMPLE

As an example, the E-R diagram for a Consultancy firm represented in KORTH’s notations \[27\] is considered. Fig. 3.9 shows the intermediate graph to get the reference graph in fig. 3.10. Fig. 3.11 shows the abstract mathematical model for the system. Table-III shows the matrix representation of the reference graph and finally from this, the mathematical model of the database schema is obtained as shown in fig. 3.13. This mathematical model is with respect to K_{SYS}. 
Figure 3.8: E-R model for a consultancy firm (Korth's notation [27])

Figure 3.9: An Intermediate Graph to get Reference Graph in fig. 3.10

[Note that the intermediate graph can also be drawn directly by analysing the context without drawing an E-R diagram]
From the reference graph in fig. 3.10, the abstract mathematical model for the database of the system is derived as shown in fig. 3.11. At this point, the decision of choosing the primary keys is pending until the actual mathematical model is developed.

**Abstract Model: Consultancy-Firm**

- **Employee**: \{SSN\}, \{KDepartment\}, NAME
- **Department**: \{D\}, \{D_NAME\}
- **Supplier**: \{S\}, \{SNAME\}
- **Part**: \{P\}, \{NAME\}
- **Sponsor**: \{SP\}, \{ACC\}, \{SP_NAME, CONTRACT\}
- **Project**: \{P\}, \{KDepartment\}, NAME
- **Dependent**: \{KEmployee\}, DEP-NO, \{SEX\}
- **Technical Employee**: \{KEmployee\}, \{RESEARCH_AREA\}
- **Administrative Employee**: \{KEmployee\}, \{TECHNICAL_QUALIFICATION\}
- **Internal Project**: \{KProject\}, \{PNAME\}
- **Funded Project**: \{KProject\}, \{SPONSOR\}, \{PNAME\}
- **Order**: \{ORDER\}, \{KSupplier, KEmployee, KPart\}, \{ORDER_DATE\}

[*: set of other differentiating attributes*]

**Figure 3.11: Abstract Mathematical Model for the Consultancy Firm**
Now, the reference graph for the database schema with respect to $K_{SYS}$ is obtained from this model as shown in fig. 3.12. The set $K_{SYS}$ contains the primary keys of the tables in the system. The DRS for the system is as follows:

\[
\begin{align*}
\text{DRS NAME} & = \text{CONSULTANCY FIRM} \\
\text{SET } K & = \{\{S\text{-SN}\}, \{S\text{-SN, DEP-NO}\}, \{D\#\}, \{P\#\}, \{ORDER\#\}, \{PT\#\}, \{SP\#\}\} \\
\text{SET EMPLOYEE} & = \{\{S\text{-SN}\}\}, \{\text{Name}\}\} \\
\text{SET DEPENDENT} & = \{\{S\text{-SN, DEP-NO}\}\}, \{\text{SEX}\}\} \\
\text{SET DEPARTMENT} & = \{\{D\#\}\}, \{D\_NAME\}\} \\
\text{SET RESEARCH_EMPLOYEE} & = \{\{S\text{-SN}\}\}, \{\text{RESEARCH AREA}\}\} \\
\text{SET TECHNICAL_EMPLOYEE} & = \{\{S\text{-SN}\}\}, \{\text{TECHNICAL QUALIFICATION}\}\} \\
\text{SET SECRETARY} & = \{\{S\text{-SN}\}\}, \{\text{PROJECT SUPERVISED}\}\} \\
\text{SET PROJECT} & = \{\{P\#\}\}, \{\text{Name}\}\} \\
\text{SET INTERNAL_PROJECT} & = \{\{P\#\}\}, \{\text{Name}\}\} \\
\text{SET FUNDED_PROJECT} & = \{\{P\#\}\}, \{\text{Name}\}\} \\
\text{SET SUPPLIER} & = \{\{S\#\}\}, \{\text{Name}\}\} \\
\text{SET PART} & = \{\{PT\#\}\}, \{\text{Name}\}\} \\
\text{SET ORDER} & = \{\{ORDER\#, \{\text{S\#}, P\#, PT\#\}\}, \{\text{S\#, P\#, PT\#, ORDER DATE}\}\} \\
\text{SET SPONSORER} & = \{\{SP\#\}\}, \{\text{ACC\#, SPNAME, CONTRACT}\}\} \\
\end{align*}
\]

\textbf{Figure 3.13: Mathematical Model for the Database Schema}

[Note that these sets will be database tables and the reference graph can be transformed to schema diagram]
Table 3.3 shows the reference matrix with respect to $K_{SYS}$ for the database of the consultancy firm.

### 3.10 Restructuring Unstructured Databases

*(A reverse engineering approach)*

Still there are organizations which maintain data in an unstructured manner having redundancies. These databases can be restructured by using this approach in a reverse way. To remove these redundancies and to restructure the databases, a reverse engineering approach is followed, where for each database table in the system a set is created. By *Equation 3.1*, it is known that there is reference from a set $A$ to another set $B$ if $A \cap B = \{b_i\}_{1 \leq i \leq m}$ and $(b_i \in B)$ & $(b_i \in K_B)$. If $\{A \cap B\}$ contains elements which do not belong to the key set of $B$ but from the describing set of $B$, then there are
redundancies. These redundancies can be removed by removing these elements, which are not in the key set, from one of the sets. If more than one attribute sets from the key set $K_B$ of the set $B$ is in $\{A \cap B\}$, then only one from $\{A \cap B\}$ must be kept in the set $A$, others should be removed from the set $A$, because it is impossible to have multiple references from $A$ to $B$.

As an example, let us consider the following database tables with lots of redundancies.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Table 2</th>
<th>Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT (ROLLNO, SNAME, ADDRESS, DOB, BLOOD_GROUP)</td>
<td>SUBJECT (SUBCODE, SUBNAME, TOTAL-MARKS, CREDITS)</td>
<td>MARKSDETAILS (ROLLNO, SNAME, SUBCODE, SUBNAME, MARKS-OBTAINED)</td>
</tr>
</tbody>
</table>

Now, these tables will be represented in the mathematical notation as follows

- $\text{SETSTUDENT} = \{\{\text{ROLLNO}\}, \{\text{SNAME, ADDRESS, DOB, BLOOD\_GROUP}\}\}$
- $\text{SETSUBJECT} = \{\{\text{SUBCODE}\}, \{\text{SUBNAME, TOTAL-MARKS, CREDITS}\}\}$
- $\text{SEIMARKSDETAILS} = \{\{\text{ROLLNO, SUBCODE}\}, \{\text{SNAME, SUBNAME, MARKS-OBTAINED}\}\}$

From this, it is obtained that

- $\text{STUDENT} \cap \text{SUBJECT} = \emptyset$ \hspace{1cm} (A)
- $\text{STUDENT} \cap \text{MARKSDETAILS} = \{\text{ROLLNO, SNAME}\}$ \hspace{1cm} (B)
- $\text{MARKSDETAILS} \cap \text{STUDENT} = \{\text{ROLLNO, SNAME}\}$ \hspace{1cm} (C)
- $\text{MARKSDETAILS} \cap \text{SUBJECT} = \{\text{SUBCODE, SUBNAME}\}$ \hspace{1cm} (D)

From B it is obtained that there is no reference from STUDENT TO MARKSDETAILS because ROLLNO and SNAME both does not belong to the key set of MARKSDETAILS. (Note that ROLLNO is not a key in MARKSDETAILS as ROLLNO and $\{\text{ROLLNO}\}$ is different.). Similarly, there is no reference from
SUBJECT to MARKSDETAILS. But there are references from MARKSDETAILS to STUDENT and SUBJECT because ROLLNO belong to the key set of STUDENT and SUBCODE belong to the key set of SUBJECT. In (C) and (D), it is observed that the intersections include attributes from the describing sets. For omission of attributes from the set to remove redundancies, consider the intersections which produce references. Here, SNAME and SUBCODE are omitted from MARKSDETAILS and the following set is obtained.

SET MARKSDETAILS = \{\{ROLLNO, SUBCODE\}, \{MARKS-OBTAINED\}\}

The corresponding E-R model will be similar to the fig. 3.2(b). The restructured database is as follows.

SET STUDENT = \{\{ROLLNO\}, \{SNAME, ADDRESS, DOB, BLOOD_GROUP\}\}

SET SUBJECT = \{\{SUBCODE\}, \{SUBNAME, TOTAL-MARKS, CREDITS\}\}

SET MARKSDETAILS = \{\{ROLLNO, SUBCODE\}, \{MARKS-OBTAINED\}\}

3.11 CONCLUSION

Though the E-R models look like graphs, the graph traversal algorithms cannot be directly applied to explore the database table references. Only the relationships among the entities can be found out. By using the reference graph, the references among the database tables can be easily explored because the reference graphs can easily be represented by using common data structures. So the process of database design can be automated with less effort. By this approach the DRS i.e. the abstract database schema can be created by using the reference matrix and the sets defined at the time of analysing the context system. In this chapter, a new data modelling approach is discussed. Also, a method for generating the database specifications is discussed which can be used as a design document in the process of software development along with
the Software Requirement Specification (SRS) document. It is a generalized format for the database schema and can be used in any DBMS.